IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Application of : **TROSS et al.**

Serial No. : 10/673,744 : Group Art Unit: 2189

Filed : September 29, 2003 : Examiner: Daniel Bokmin Ko

For : LOW-COST REMOTE DATA MIRRORING

Honorable Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

DECLARATION UNDER 37 CFR 1.131

Sir:

We, the undersigned, Martin Tross and Aviad Zlotnick, hereby declare as follows:

1) We are the Applicants in U.S. Patent Application No. 10/673,744 (hereinafter "the Application"), and are the inventors of the subject matter described and claimed in claims 1-60 therein.

2) Prior to March 21, 2003, we conceived our invention, as described and claimed in the Application, in Israel, a WTO country. Prior conception of the invention is evidenced by IBM Disclosure IL8-2003-0016, which was prepared by Aviad Zlotnick prior to March 21, 2003, and was submitted to the IBM disclosure database on April 1, 2003. A copy of the disclosure record is attached hereto as Exhibit A, and the disclosure itself is attached as Exhibit B. The date that is blacked out

in Exhibit A is prior to March 21, 2003.

3) The following table shows the correspondence between the elements of claim 1 in the present patent application and elements of the disclosure in Exhibit B. The elements of the claim are embodied mainly in paragraph 4 of page 2 in the exhibit, referring to "Writes on control unit A." This paragraph presents, in tabular form, the essential elements of an algorithm used in carrying out the method of claim 1, wherein "control unit A" refers variously to the primary and secondary storage controllers (as indicated explicitly in line 1 of the algorithm).

Claim 1	**Exhibit B**
A method for storing data in a data storage system that includes primary and secondary storage subsystems, including respective first and second volatile cache memories and respective first and second non-volatile storage media	Page 1, paragraph 4: "The essence of this invention is to use the [cache] memory of the remote [secondary] controller as the backup of the [cache] memory on the primary controller." It was well known at the time we filed this application that data storage mirroring systems comprise primary and secondary storage subsystems with respective non-volatile storage media. The terms "cache" and "memory" are used interchangeably in the disclosure.

Claim 1	**Exhibit B**
receiving the data at the primary storage subsystem from a host processor	Page 2, paragraph 4 (referring to "Writes on control unit A"): "1. Accept a write… (If A is a primary the write is from a host…)"
writing the data to the first volatile cache memory in the primary storage subsystem	Continuation of preceding item: "update the local cache."
copying the data from the primary storage subsystem to the secondary storage subsystem	Page 2, paragraph 4: "3. If A is a primary, write the data to the secondary…"
writing the copied data to the second volatile cache memory in the secondary storage subsystem	Page 2, paragraph 4: "1. Accept a write, update the local cache. (… If A is a secondary, the write is from the primary…)"

Claim 1	**Exhibit B**
returning an acknowledgment from the secondary storage subsystem to the primary storage subsystem responsively to writing the copied data to the second volatile cache memory and prior to saving the data in the second non-volatile storage media	Page 2, paragraph 4, taking "A" to be the secondary: "4. Acknowledge the original write," i.e., send the acknowledgment to the "primary B" after updating the local cache and marking the written tracks in a "change recording bitmap." Writing the data to the non-volatile storage media of the secondary is performed subsequently in the "periodic cache destage" (page 2, paragraph 5).
signaling the host processor that the data have been stored in the data storage system responsively to the acknowledgment from the secondary storage subsystem	Page 2, paragraph 4: "3. If A is a primary, write the data to the secondary, and wait for an acknowledgment." When the acknowledgment from the secondary is received, the primary will then "4. Acknowledge the original write."
transferring the data in the primary and secondary storage subsystems from the first and second volatile cache memories to the first and second non-volatile storage media, respectively	Page 2, paragraph 5: "Periodic cache destage scan on A." "Destaging" is a term of art that means transferring data from cache to non-volatile memory. As noted above, "control unit A" refers variously to the primary and secondary storage controllers.

4) Independent claims 21 and 41 recite a data storage system and a computer software product, which operate on principles similar to those of method claim 1. Based on the similarity of subject matter between the method, system and software claims, it can similarly be demonstrated that the elements of claims 21 and 41 are described in Exhibit B.

5) In accordance with standard procedures at IBM, the disclosure (Exhibit B) was sent out for review by three members of the IBM technical staff (Gail Spear, Rob Nicholson and Dalit Tzafrir). They submitted their evaluations on April 13, April 25 and April 29, 2003, respectively. The evaluation reports are attached hereto as Exhibit C.

6) Following discussion of these evaluations, we felt that it would be advantageous to seek further input from Tom Weaver, who was at the time one of the architects of the IBM HAGEO storage mirroring system. We received Mr. Weaver's response on June 2, 2003. An e-mail exchange reporting this response is attached hereto as Exhibit D. (Note that dates in this and other exhibits are presented in the form "day/month/year," following international convention.)

7) Our invention was subsequently brought up for discussion in the June meeting of the invention review forum at the IBM Haifa Research Laboratory, where it was decided that a patent application covering this invention should be filed. An e-mail from the Intellectual

Property Department informing us of this decision is attached hereto as Exhibit E.

8) On July 10, 2003, Aviad Zlotnick met with Dr. Daniel Kligler, of Sanford T. Colb & Co., who was retained by IBM as outside counsel for the purpose of preparing this patent application. Immediately following the meeting, Dr. Kligler sent a memo to Suzanne Erez, IP Manager of the IBM Haifa Research Laboratory, summarizing the meeting and timetable for completion of a draft of the application. A copy of this memo is attached hereto as Exhibit F.

9) On July 17, 2003, Dr. Kligler sent us a first draft of the patent application. A copy of the draft with Dr. Kligler's cover letter is attached hereto as Exhibit G.

10) We reviewed the first draft and concurrently did further work to fill in details of alternative embodiments of our invention. We sent our comments on the draft to Dr. Kligler on August 10, 2003. A copy of our cover letter to Dr. Kligler is attached hereto as Exhibit H.

11) On August 13, 2003, Dr. Kligler sent us a revised draft of the patent application. A copy of Dr. Kligler's cover letter is attached hereto as Exhibit I. The draft itself was similar to the patent application that was subsequently filed.

12) We sent Dr. Kligler our comments on the revised

draft in a letter dated August 27, 2003, which is attached hereto as Exhibit J.

13) On August 31, 2003, Dr. Kligler sent us the final draft of the patent application. A copy of Dr. Kligler's cover letter to us is attached hereto as Exhibit K.

14) We immediately gave our approval of this draft. Dr. Kligler sent the text and figures to Ms. Erez on September 1, 2003. A copy of Dr. Kligler's cover letter is attached hereto as Exhibit L.

15) Ms. Erez sent the completed application to an attorney in the IBM Watson Research Center for filing. She received a letter back from Michelle Parra, a paralegal in the Watson Research Center, on September 11, 2003, indicating that the formal drawings prepared by Dr. Kligler's firm were not in the proper format for filing. Ms. Erez passed this message along to Dr. Kligler on September 14, 2003, with a request to revise the drawings. A copy of this correspondence is attached hereto as Exhibit M. Dr. Kligler's draftsman revised the drawings as requested, and the application was then filed on September 29, 2003.

16) Thus, to summarize, the facts set forth above and supported by Exhibits A-M demonstrate that prior to March 21, 2003, we conceived the invention recited in the claims of the Application and were then diligent in constructive reduction to practice of the invention during the period between March 21 and September 29, 2003.

We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and conjecture are thought to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application of any patent issued thereon.

Martin Tross, Citizen of Israel — Date
5 Geulei Teiman Street, Haifa 34991, Israel

Aviad Zlotnick, Citizen of Israel — Aug. 1, '06
Date
Mizpe Netofa, D.N. Galil Takhton 15295, Israel

We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and conjecture are thought to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application of any patent issued thereon.

Martin Tross, Citizen of Israel — Date: 1 August 2006
5 Geulei Teiman Street, Haifa 34991, Israel

Aviad Zlotnick, Citizen of Israel — Date
Mizpe Netofa, D.N. Galil Takhton 15295, Israel

EXHIBIT A

Disclosure IL 8-2003-0016
Prepared for and/or by an IBM Attorney - IBM Confidential

Created By Aviad Zlotnick On [redacted] 03:17:15 PM ZE2
Last Modified By Enterprise Agentmgr On 09/10/2004 09:32:58 PM EST
Archived on 08/10/2004

Required fields are marked with the asterisk (*) and must be filled in to complete the form.

***Title of disclosure (in English)**
Low Cost PPRC

Summary

Status	Final Decision (File)
Final deadline	
Final deadline reason	
Docket family	IL9-2003-0033
* Processing location	Israel
* Functional area	(11) Storage & Systems Technology
Attorney/Patent professional	Suzanne Erez/Haifa/IBM
Invention development team (IDT)	Gail Spear/Tucson/IBM Rob Nicholson/UK/IBM Dalit Tzafrir/Haifa/IBM Alain Azagury/Haifa/IBM Suzanne Erez/Haifa/IBM Michael Rodeh/Haifa/IBM Gal Ashour/Haifa/IBM
Submitted date	01/04/2003 10:41:01 AM CEDT
* Owning division	HRL
Incentive program	
Lab	
* Technology code	344
Patent value tool (PVT) score	27

Inventors with a Blue Pages entry

Inventors: Aviad Zlotnick/Haifa/IBM, Martin Tross/Haifa/IBM

Inventor Name	Inventor Serial	Div/Dept	Inventor Phone	Manager Name
Zlotnick, Aviad	948501	N/A/62170	N/A	Tross, Martin
Tross, Martin	721800	5	N/A	Azagury, Alain
		N/A/62170		
		5		

> denotes primary contact

Inventors without a Blue Pages entry

Invention Development Team Information
Main Idea

To view the Main Idea of this disclosure, open the "Main Idea" document from the view
***Inventor Questions**
Evaluation
Final Decision

Main Idea for Disclosure IL 8-2003-0016

Prepared for and/or by an IBM Attorney - IBM Confidential

Archived On 10/06/2003 08:03:52 AM

Title of disclosure (in English)
Low Cost PPRC

Main Idea

1. Background: What is the problem solved by your invention? Describe known solutions to this problem (if any). What are the drawbacks of such known solutions, or why is an additional solution required? Cite any relevant technical documents or references.
Storage controllers usually combine a fast none persistent memory of limited size (the cache) and a slower persistent memory with capacity to support all of the data (disk, raid). In order to guarantee that data in the cache which has not yet been copied to disk is not lost in case of loss of power or crash of the controller, controllers often use a second memory, (sometimes in a separate node of controller, [Lodestone,Shark]) . The controllers have a mechanism for keeping track of the data which has been copied to disk, and releasing the space occupied by this data in the second (also the first) memory. In the Shark the second memory is persistent, slower and much more expensive than the regular cache. In continuous mirroring a copy of the data is also maintained on a storage controller in a remote site, so in practice one finds up to four cache copies of the same data.

This invention disclosure describes a mirroring solution that does not require special hardware, uses at most two cached copies of the data ,and yet is still immune to a Single Point of Failure in terms of data usability. Combined with a system that ensures processing continuity, such as HACMP, continuous operation is also ensured over a single point of failure. Although they do not provide as much robustness as the Shark Enterprise Storage Systems, such systems may be attractive because of their significantly lower cost. Furthermore, two mirrored controllers in the same site may also be considered a single, inexpensive, but robust storage control unit.

Current mirroring solutions are based on having the primary control unit keeping track of data that are yet to be transferred to the secondary, and the secondary keeping track of data that are yet to be hardened. In case of a failure on the primary, or failure of communications, after recovery the unsent data will be transferred to the secondary. In case of failure in the secondary, unhardened data will be requested from the primary. As mentioned above, this solution needs special hardware, and is therefore expensive.

2. Summary of Invention: Briefly describe the core idea of your invention (saving the details for questions #3 below). Describe the advantage(s) of using your invention instead of the known solutions described above.
The essence of this invention is to use the memory of the remote controller as the back up of the memory on the primary controller, and vice versa. In addition, each controller has a mechanism for knowing which data is in cache on the other controller and may not have been written to disk yet.

We propose to define (actually, make explicit) the following mirroring requirement. The amount of data that has to be re-transferred to the secondary, or copied from the secondary back to the primary when recovering from a failure will be limited and controllable. This requirement reflects the need to ensure that returning to fully mirrored operation after a failure does not take too long.

To achieve this goal, we propose to have each control unit keep track of the written data that may not have been hardened on the other control unit. When one of the control unit fails and recovers, the other control unit knows which data may need to updated.

If a primary fails, the secondary can be instructed to be a primary and accept writes from hosts. Read operations can be directed to either of the CUs at any time

3. Description: Describe how your invention works, and how it could be implemented, using text, diagrams and flow charts as appropriate.
The flow of control is the same on both control units, and consists of several operations:

(The description here uses a change recording bitmap to keep track of modified data, but many other data structures could be used with out impacting the algorithm)

Writes on control unit A:
1. Accept a write, update the local cache. (If A is a primary the write is from a host. If A is a secondary the write is from the primary B).
2. Mark the written track(s) in a change recording bitmap
3. If A is a primary, write the data to the secondary, and wait for an acknowledgement.
4. Acknowledge the original write.

Periodic cache destage scan on A:
1. Send a message to B that the destage scan is starting
2. Wait for acknowledgement
3. Perform the destage scan
4. Send a message to B that the destage scan has ended

Change recording management on A:
1. When B starts a destage scan start a new change recording bitmap, and preserve the previous on as "the old CR bitmap". Any previous old CR bitmap is discarded.
2. When B finishes the destage scan clear the old CR bitmap.

Operations on A for Recovery on B:
1. OR the current CR bitmap with the old one.
2. Write to B all the data that is marked in the CR bitmap.

Operations on A for recovery of A (assume A crashed and needs to be brought up-to-date w/o server failover
1. After IML request B to send all data indicated by both CR maps on B
2. wait for B to inform A that all data sent
3. Resume operation

One can do without a destage scan using the following alternative:
- A counter N of staged tracks is defined.
- The cache destage operation increments N whenever it performs a destage of a track that is marked in the old CR bitmap. The old CR bit is reset when its track is destaged
- N is set to 0 whenever a new CR bitmap is created.
- Immediately after setting N to 0, count of the dirty tracks that are marked in the old CR bitmap, and put the count in M.
- Use a lock to make sure the destage and the count process are mutually exclusive per old CR bit.
- When N and M are equal there are no more dirty bits in the old CR bitmap. This is the equivalent if an end of destage scan.

Yet another alternative is to scan the old CR bitmap and request a synchronous destage of any track that has a CR bit set and is dirty in cache.

Dividing the CR bitmap to small areas, and treating each as a separate cache may improve performance significantly.

IP&L Disclosure Evaluation : IL8-2003-0016

Prepared for and/or by an IBM Attorney - IBM Confidential

Created By Dalit Tzafrir On 29/04/2003 08:15:29 AM EDT
Last Modified By Enterprise Agentmgr On 09/10/2004 09:32:58 PM EST
Archived on 08/10/2004

Required fields are marked with the asterisk (*) and must be filled in to complete the form .

Title of disclosure
Low Cost PPRC

Date evaluation due to IPL: 08/05/2003	Date evaluation submitted to IP&L: 29/04/2003

A. Threshold Questions

* 1. Operability - Is implementation of the invention possible?
- ● Yes
- ○ No
- ○ More information required

Reasons for above answer:

* 2. Novelty- Are one or more concept(s) of the invention novel over what is already known in the literature, existing commercial products, patents, and earlier IBM invention disclosures?
- ● Yes
- ○ No

Reasons for above answer:

B. Valuation Questions

* 1. Adequacy of description:
- ○ Inadequate; invention unclear from description
- ○ Incomplete; essential features missing
- ○ Further clarification or implementation detail needed
- ● Clear and complete as is

State reason for answer:

* 2. Technical contribution of invention:
- ○ None
- ○ Minor addition to known technology
- ● Significant addition to known technology
- ○ Major advance in technology

Reasons for above answer:

* 3. Describe the problem solved/benefit provided and the implementation cost of the invention compared to existing or reasonably expected alternatives:
- ○ Minor problem/incremental benefit - significant implementation cost
- ● Significant problem; substantial benefit - significant implementation cost
- ○ Minor problem/incremental benefit - minor implementation cost
- ○ Significant problem/substantial benefit - minor implementation cost

* **4.** Are any alternatives to the invention available to those wishing to avoid its use?
- ○ Suitable alternatives available
- ● Alternatives have drawbacks
- ○ No feasible alternatives

* **5.** Describe the likelihood of use of the invention (answer each):

IBM's customers?	○ Unlikely	● Possible	○ Probable	○ Definite
IBM's suppliers/vendors?	● Unlikely	○ Possible	○ Probable	○ Definite
IBM's competitors?	○ Unlikely	● Possible	○ Probable	○ Definite
IBM?	○ Unlikely	● Possible	○ Probable	○ Definite

Reasons for above answer:

* **6.** What % of third party products in the technical field will likely contain the invention?
- ● < 25%
- ○ 25-50%
- ○ 50-75%
- ○ > 75%

* **7.** How long is the invention likely to be used in products by IBM or others?
- ○ < 5 years
- ● 5-10 years
- ○ 10-15 years
- ○ > 15 years

* **8.** How easily can use of the invention by a third party be detected?
- ○ Undiscoverable; third party must admit use for IBM to know
- ● Difficult; e.g.; with reverse engineering or examination of available code
- ○ With work; e.g.; using test cases; but not reverse engineering
- ○ Easily; by running & viewing product operation
- ○ Trivially; without purchase of product; e.g.; by reading product literature

Reasons for the above answer, including description of how use could be detected:

Evaluator recommended decision : ○ Close
○ Publish
● Search

Close: A patent would probably have little licensing value or IBM's freedom of use is already assured or is not important

Publish: A patent would probably have limited licensing value to IBM but freedom of use should be preserved

Search: A patent on this subject could have significant licensing value. IPLaw should provide an opinion on patentability and portfolio value and a recommendation whether to file a patent application

☐ Additional Search Info: This disclosure should be MERGED before searching and filing with disclosure(s)

Comments (Note: Limit your comments to technical /business issues)

Form Revised (05/28/03)

IP&L Disclosure Evaluation : IL8-2003-0016
Prepared for and/or by an IBM Attorney - IBM Confidential

Created By Rob Nicholson On 25/04/2003 09:54:50 AM GDT
Last Modified By Enterprise Agentmgr On 09/10/2004 09:32:59 PM EST
Archived on 08/10/2004

Required fields are marked with the asterisk (*) and must be filled in to complete the form .

Title of disclosure
Low Cost PPRC

Date evaluation due to IPL: 08/05/2003	Date evaluation submitted to IP&L: 25/04/2003

A. Threshold Questions

* **1. Operability** - Is implementation of the invention possible?
- ● Yes
- ○ No
- ○ More information required

Reasons for above answer:
It is clear from reading the disclosure and its referneced patent application that the invention could be implemented as described and would provide the benefits claimed.

* **2. Novelty**- Are one or more concept(s) of the invention novel over what is already known in the literature, existing commercial products, patents, and earlier IBM invention disclosures?
- ● Yes
- ○ No

Reasons for above answer:
whilst the overall concepts described are not novel the application of the concepts to journalling metadata in a storage controller are novel.

B. Valuation Questions

* **1.** Adequacy of description:
- ○ Inadequate; invention unclear from description
- ○ Incomplete; essential features missing
- ○ Further clarification or implementation detail needed
- ○ Clear and complete as is

State reason for answer:
Unlike the other evaluators I am not familialr with the internal design of the Shark. I needed to carefully read US patent application US2002/0083263 A1 in order to understand this invention. This is the patent application which resulted from the referenced IL920000019.

* **2.** Technical contribution of invention:
- ○ None
- ● Minor addition to known technology
- ○ Significant addition to known technology
- ○ Major advance in technology

Reasons for above answer:
This is an enhancement to an existing invention. It is unlikely to be useful unless the invention described in
US2002/0083263 A1 is also being used.

* **3.** Describe the problem solved/benefit provided and the implementation cost of the invention compared to existing or reasonably expected alternatives:

○ Minor problem/incremental benefit - significant implementation cost
○ Significant problem; substantial benefit - significant implementation cost
● Minor problem/incremental benefit - minor implementation cost
○ Significant problem/substantial benefit - minor implementation cost

* **4.** Are any alternatives to the invention available to those wishing to avoid its use?
○ Suitable alternatives available
● Alternatives have drawbacks
○ No feasible alternatives

* **5.** Describe the likelihood of use of the invention (answer each):

IBM's customers?	○ Unlikely	○ Possible	○ Probable	● Definite
IBM's suppliers/vendors?	● Unlikely	○ Possible	○ Probable	○ Definite
IBM's competitors?	○ Unlikely	● Possible	○ Probable	○ Definite
IBM?	○ Unlikely	○ Possible	○ Probable	● Definite

Reasons for above answer:
this is only likely to be used by competitors if they have a similar architecture to that of ESS

* **6.** What % of third party products in the technical field will likely contain the invention?
● < 25%
○ 25-50%
○ 50-75%
○ > 75%

* **7.** How long is the invention likely to be used in products by IBM or others?
○ < 5 years
○ 5-10 years
● 10-15 years
○ > 15 years

* **8.** How easily can use of the invention by a third party be detected?
○ Undiscoverable; third party must admit use for IBM to know
● Difficult; e.g.; with reverse engineering or examination of available code
○ With work; e.g.; using test cases; but not reverse engineering
○ Easily; by running & viewing product operation
○ Trivially; without purchase of product; e.g.; by reading product literature
Reasons for the above answer, including description of how use could be detected:

Evaluator recommended decision : ○ Close
○ Publish
● Search

Close: A patent would probably have little licensing value or IBM's freedom of use is already assured or is not important

Publish: A patent would probably have limited licensing value to IBM but freedom of use should be preserved

Search: A patent on this subject could have significant licensing value. IPLaw should provide an opinion on patentability and portfolio value and a recommendation whether to file a patent application

☐ Additional Search Info: This disclosure should be MERGED before searching and filing with disclosure(s)

Comments (Note: Limit your comments to technical /business issues)

Form Revised (05/28/03)

IP&L Disclosure Evaluation : IL8-2003-0016
Prepared for and/or by an IBM Attorney - IBM Confidential

Created By Gail Spear On 13/04/2003 01:58:47 AM CEDT
Last Modified By Enterprise Agentmgr On 09/10/2004 09:32:59 PM EST
Archived on 08/10/2004

Required fields are marked with the asterisk (*) and must be filled in to complete the form.

Title of disclosure
Low Cost PPRC

Date evaluation due to IPL: 08/05/2003	Date evaluation submitted to IP&L: 12/04/2003

A. Threshold Questions

* 1. Operability - Is implementation of the invention possible?

- ●
- ○
- ○

Reasons for above answer:

* 2. Novelty- Are one or more concept(s) of the invention novel over what is already known in the literature, existing commercial products, patents, and earlier IBM invention disclosures?

- ●
- ○

Reasons for above answer:

using a remote controller as the second copy of the memory is unique.

B. Valuation Questions

* 1. Adequacy of description:

- ○
- ○
- ○
- ●

State reason for answer:

* 2. Technical contribution of invention:

- ○
- ○
- ●
- ○

Reasons for above answer:

* 3. Describe the problem solved/benefit provided and the implementation cost of the invention compared to existing or reasonably expected alternatives:

* **4.** Are any alternatives to the invention available to those wishing to avoid its use?

* **5.** Describe the likelihood of use of the invention (answer each):

IBM's customers?	●	○	○	○
IBM's suppliers/vendors?	●	○	○	○
IBM's competitors?	○	●	○	○
IBM?	○	●	○	○

Reasons for above answer:
IBM may use this for a low-cost storage controller. IBM's competitors might also like to use this if they find out about it.

* **6.** What % of third party products in the technical field will likely contain the invention?

* **7.** How long is the invention likely to be used in products by IBM or others?

* **8.** How easily can use of the invention by a third party be detected?

Reasons for the above answer, including description of how use could be detected:

Evaluator recommended decision : ○ Close
○ Publish
● Search

Close: A patent would probably have little licensing value or IBM's freedom of use is already assured or is not important

Publish: A patent would probably have limited licensing value to IBM but freedom of use should be preserved

Search: A patent on this subject could have significant licensing value. IPLaw should provide an opinion on patentability and portfolio value and a recommendation whether to file a patent application

☐ Additional Search Info: This disclosure should be MERGED before searching and filing with disclosure(s)

Comments (Note: Limit your comments to technical /business issues)
this evaluation replaces my evaluation dated April 9,2003

EXHIBIT D

Alain Azagury
02/06/2003 11:12 PM
This document expires on 31/08/2003

To: Suzanne Erez/Haifa/IBM@IBMIL
cc:
From: Alain Azagury/Haifa/IBM@IBMIL
Subject: Fw: Fast Synchronous Mirroring -- (Low Cost PPRC, IL8-2003-0016)

Suzanne,

Pls refresh my memory. What did we resolve for this one?

Alain
---- Forwarded by Alain Azagury/Haifa/IBM on 02/06/2003 11:11 PM ----

Aviad Zlotnick
02/06/2003 03:22 PM

To: Alain Azagury/Haifa/IBM
cc: Suzanne Erez/Haifa/IBM, Martin Tross/Haifa/IBM, Michael Factor/Haifa/IBM
From: Aviad Zlotnick/Haifa/IBM@IBMIL
Subject: Fw: Fast Synchronous Mirroring -- (Low Cost PPRC, IL8-2003-0016)

Alain,

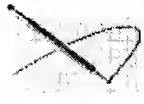
Seeing that the Low Cost PPRC disclosure was not an immediate "Search", Martin and I called Tom Weaver, one of the architects in the HA group. He was very positive, I dare say enthused, about this technology. I am forwarding his note.

Cheers,

Aviad

Tel: +972-48-296-284, Cell: +972-53-742-585, Fax: +972-48-296-112, Email: aviad@il.ibm.com

---- Forwarded by Aviad Zlotnick/Haifa/IBM on 02/06/03 03:14 PM ----



Thomas Weaver@IBMUS
02/06/03 01:54 PM

To: Aviad Zlotnick/Haifa/IBM@IBMIL@IBMDE
cc:
From: Thomas Weaver/Austin/IBM@IBMUS
Subject: Re: Fast Syncronous Mirroring

Avaid,

I believe that the technology you describe would be a useful addition to the current HAGEO product. As we plan future work on HAGEO, I will try to have it incorporated in the product. (Actual timing of that effort depends on other business concerns).

I also think that this technology should be submitted for a patent search.

Tom Weaver
tvweaver@austin.ibm.com

EXHIBIT E

Hadas Cohen Bar -Gil
06/07/2003 10:44 AM
This document expires on 04/10/2003

To: Aviad Zlotnick/Haifa/IBM@IBMIL, Martin Tross/Haifa/IBM@IBMIL
cc: Alain Azagury/Haifa/IBM@IBMIL, HRL IP Department/Haifa/IBM@IBMIL
Subject: *IBM Confidential: Final Decision for IL820030016
Importance:

DISCLOSURE NUMBER: IL820030016
DOCKET NUMBER: IL920030033
TITLE: Low Cost PPRC

We are pleased to inform you that your third level manager has decided to FILE a patent application on the invention described in your invention disclosure.

The IP department will contact you in the near future to discuss the preparation of the patent application and the meeting with the outside counsel, at which time we may request additional information regarding this invention.
IMPORTANT: Be sure to identify ALL related prior information you are aware of that is MATERIAL to this invention. Failure to do so can result in the invalidity of any patent issuing on your invention.
Please verify the information in your inventor profile located in the World Wide Patent Tracking System to facilitate the filing of your patent application. Please refer to the above referenced Docket Number for future inquiries regarding this invention.
IBM appreciates your time and efforts in submitting this disclosure and the creative thought that it represents.
Thank you again for your assistance in protecting IBM's intellectual property.

Regards,

Hadas Cohen Bar-Gil,
Intellectual Property Department
IBM Haifa Laboratory
Phone: +972-4-829-6145
Fax: +972-4-829-6521
Email: HRL IP Department/Haifa/IBM,

IBM CONFIDENTIAL

Date: July 11, 2003

To: Suzanne Erez, IBM

From: Daniel Kligler
Sanford T. Colb & Co.

Re: IL9-2003-0033 – our ref. 49267 - estimate of time and charges

Title: Low-cost PPRC

Inventors: Martin Tross, Aviad Zlotnick

Meeting held: July 10, 2003

Materials received: Invention disclosure, background article

Time est.: First draft to be completed by early August

Cost est.: $6,000 + VAT in professional fees, not including filing costs or out-of-pocket expenses.

Comments: Aviad has not done a patent search and is not aware of any relevant art. He said that you may have done or plan to do a search of the patent database. If so, please send me any references you find.

EXHIBIT G

From: Daniel Kligler
To: Aviad <aviad@il.ibm.com>
Date: 7/17/03 11:47AM
Subject: Low-cost PPRC - IL9-2003-0033, our ref. 49267 - IBM CONFIDENTIAL

Dear Aviad,

Attached please find a first draft of this application (text and figures).

Please review this draft, and let us have your corrections and comments at your earliest opportunity. Note a number of questions to you that I have marked in boldface in the text.

Regards,
Danny

P.S. to Suzanne: Although the learning curve on this application was a little steeper than I anticipated, writing it went pretty quickly, and I expect that Aviad's other applications will likewise be ready earlier than the dates I wrote in my memos to you last week.

>>> "HRL IP Department" <HRL_IPDEPARTMENT@il.ibm.com> 07/06/03 10:40AM >>>
Hello Daniel,

Enclosed please find the main ideas for the four new applications:

IL9-2003-0028 by:
Hayardeny Amiram
Teperman Avi
Tross Martin
Zlotnick Aviad
(See attached file: Main Idea IL9-2003-0028.doc)

IL9-2003-0031 by:
Hayardeny Amiram
Tross Martin
Zlotnick Aviad
(See attached file: Main Idea IL9-2003-0031.doc)

IL9-2003-0032 by:
Fienblit Shachar
Tross Martin
Zlotnick Aviad
(See attached file: Main Idea IL9-2003-0032.doc)

IL9-2003-0033 by:
Tross Martin
Zlotnick Aviad
(See attached file: Main Idea IL9-2003-0033.doc)

If any further material/information is needed, please let me know.

Have a nice day,

Hadas Cohen Bar-Gil,
Intellectual Property Department
Operation Services, IBM HRL
Volce: +972-4-829-6145 Fax: +972-4-829-6521

CC: Department, HRL IP; Erez, Suzanne

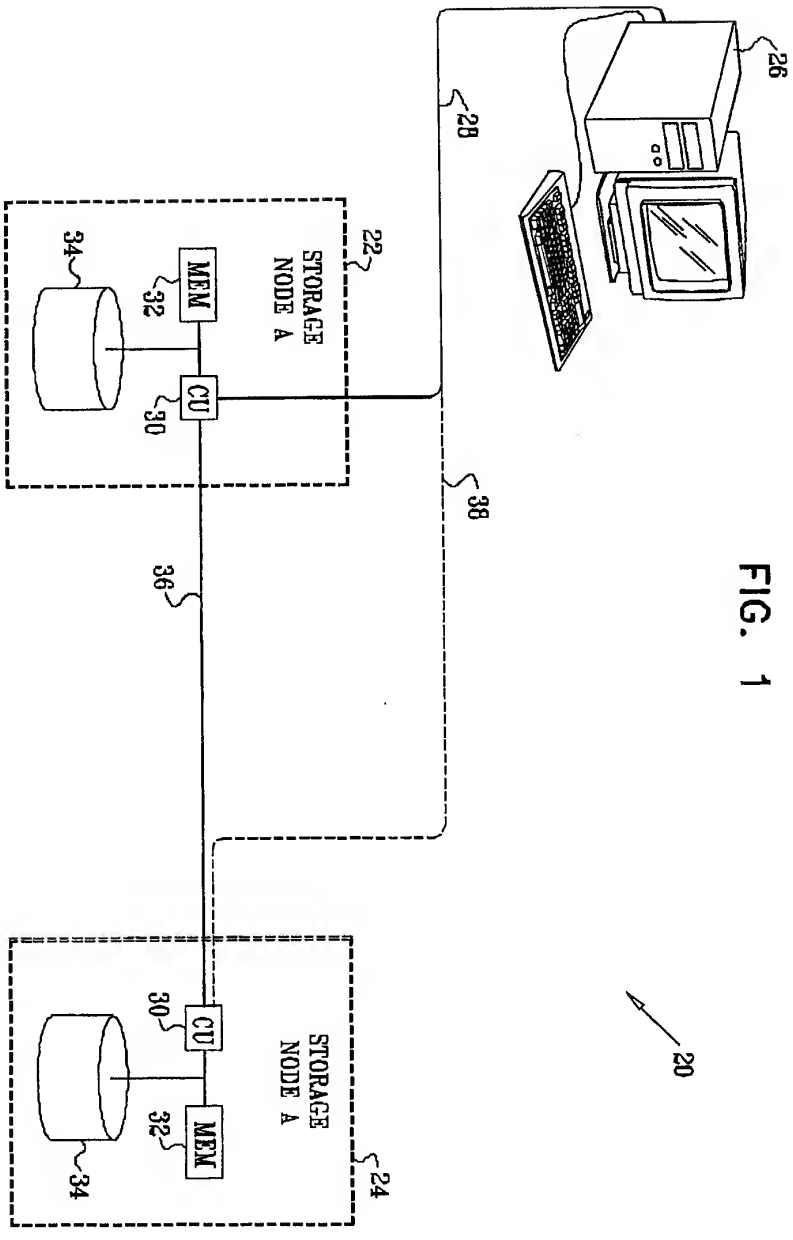

26
28
STORAGE NODE A
22
34
MEM
32
CU
30
38
36
30
CU
STORAGE NODE A
24
32
MEM
34
20

FIG. 1

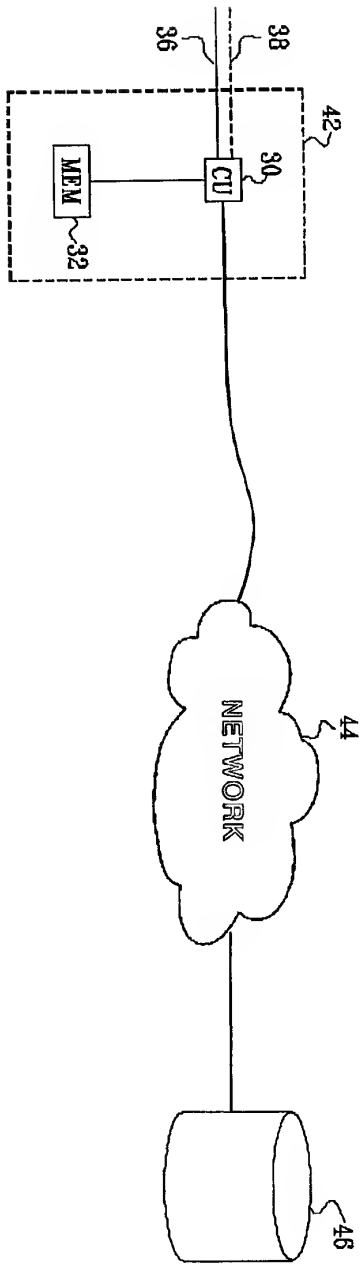


FIG. 2

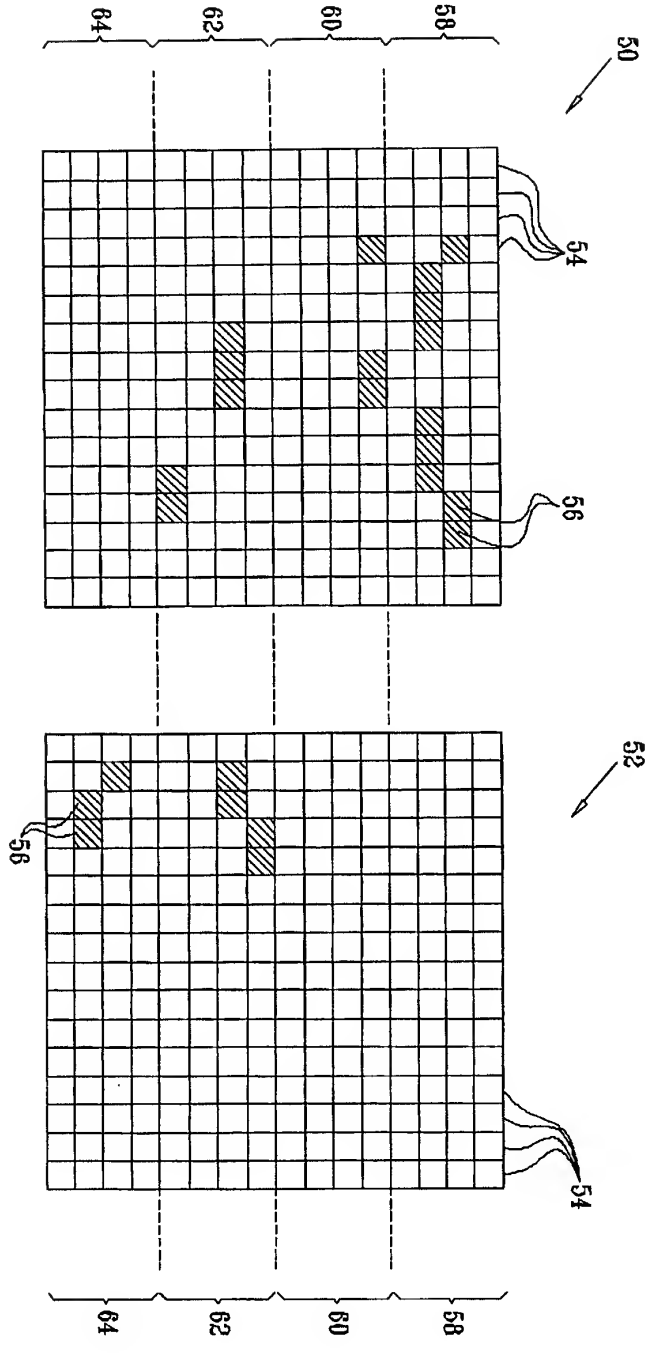


FIG. 3

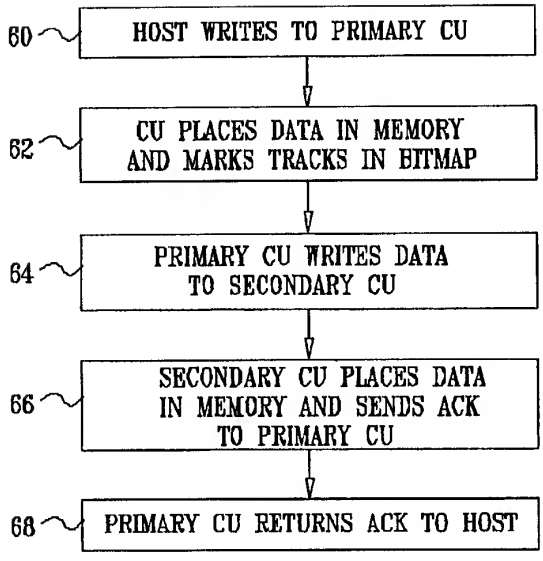
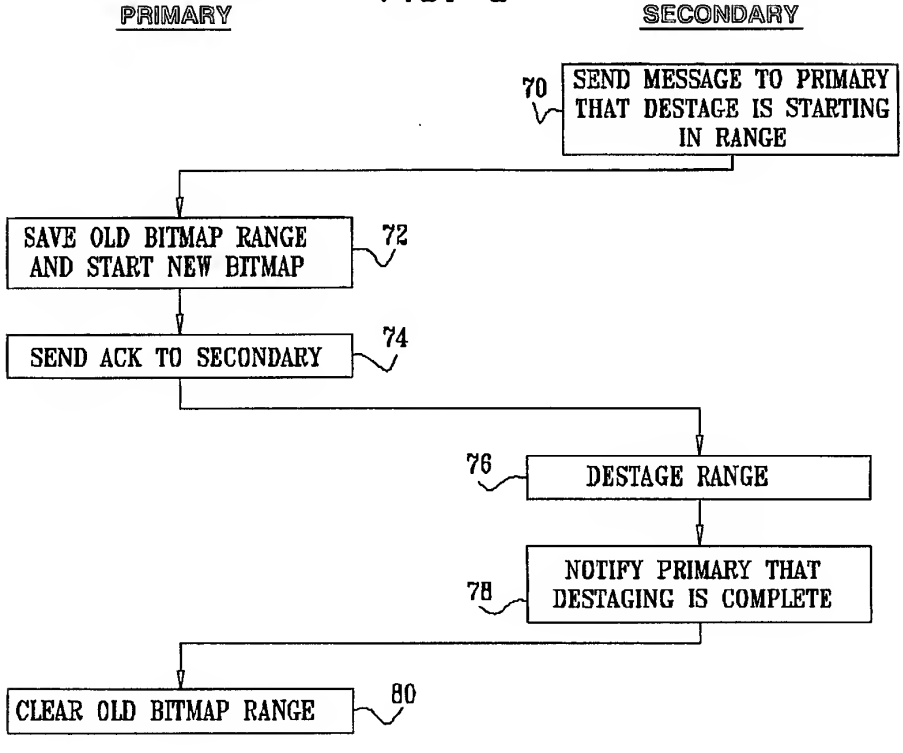

60
HOST WRITES TO PRIMARY CU
62
CU PLACES DATA IN MEMORY AND MARKS TRACKS IN BITMAP
64
PRIMARY CU WRITES DATA TO SECONDARY CU
66
SECONDARY CU PLACES DATA IN MEMORY AND SENDS ACK TO PRIMARY CU
68
PRIMARY CU RETURNS ACK TO HOST

FIG. 4

FIG. 5


PRIMARY
SECONDARY
70
SEND MESSAGE TO PRIMARY THAT DESTAGE IS STARTING IN RANGE
SAVE OLD BITMAP RANGE AND START NEW BITMAP
72
SEND ACK TO SECONDARY
74
76
DESTAGE RANGE
78
NOTIFY PRIMARY THAT DESTAGING IS COMPLETE
CLEAR OLD BITMAP RANGE
80

LOW-COST REMOTE DATA MIRRORING

FIELD OF THE INVENTION

The present invention relates generally to data storage systems, and specifically to data mirroring for failure protection in storage systems.

BACKGROUND OF THE INVENTION

Data backup is a standard part of all large-scale computer data storage systems (and most small systems, as well). Data written to a primary storage medium, such as a volume on a local storage subsystem, are copied, or "mirrored," to a backup medium, typically another volume on a remote storage subsystem. The backup volume can then be used for recovery in case a disaster causes the data on the primary medium to be lost. Methods of remote data mirroring are surveyed by Ji et al., in an article entitled "Seneca: Remote Mirroring Done Write," *Proceedings of USENIX Technical Conference* (San Antonio, Texas, June, 2003), pages 253-268, which is incorporated herein by reference. The authors note that design choices for remote mirroring must attempt to satisfy the competing goals of keeping copies as closely synchronized as possible, while delaying foreground writes by host processors to the local storage subsystem as little as possible.

Large-scale storage systems, such as the IBM Enterprise Storage Server (ESS) (IBM Corporation, Armonk, New York), typically offer a number of different copy service functions that can be used for remote mirroring. Among these functions is peer-to-peer remote copy (PPRC), in which a mirror copy of a source volume on a primary

storage subsystem is created on a secondary storage subsystem. When an application on a host processor writes to a PPRC volume on the primary subsystem, the corresponding data updates are entered into cache memory and non-volatile storage at the primary subsystem. The control unit (CU) of the primary subsystem then sends the updates over a communication link to the secondary subsystem. When the CU of the secondary subsystem has placed the data in its own cache and non-volatile storage, it acknowledges receipt of the data. The primary subsystem then signals the application that the write operation is complete.

PPRC provides host applications with essentially complete security against single-point failures, since all data are written synchronously to non-volatile media in both the primary and secondary storage subsystems. On the other hand, the need to save all data in non-volatile storage on both subsystems before the host write operation is considered complete can introduce substantial latency into host write operations. In some large-scale storage systems, such as the above-mentioned IBM ESS, this latency is reduced by initially writing data both to cache and to high-speed, non-volatile media, such as non-volatile random access memory (RAM), in both the primary and secondary subsystems. The data are subsequently copied to disk asynchronously (an operation that is also referred to as "hardening" the data) and removed from the non-volatile memory. The large amount of non-volatile memory that must be used for this purpose is very costly.

SUMMARY OF THE INVENTION

The present invention provides methods for data mirroring that can be used to create storage systems that are immune to single-point failures and have low-latency write response, without requiring special non-volatile memory or other costly components. In embodiments of the present invention, when a host writes data to a primary storage subsystem, the primary storage subsystem records the data in volatile cache memory, and transmits a copy of the data to the secondary storage subsystem. The secondary storage subsystem likewise writes the data to its cache, and sends an immediate acknowledgment to the primary storage subsystem. The primary storage subsystem then signals the host to acknowledge that the write operation has been completed, without waiting for the data to be written to the disk (or other non-volatile media) on either the primary or secondary storage subsystem.

Both primary and secondary storage subsystems keep a record of the address ranges of data that the other subsystem has received in its cache, but may not yet have copied to non-volatile storage. In the event of a failure in one of the subsystems, this record indicates which data will have to be copied back to the failed subsystem during recovery (in addition to any new data that may have been written to the operating subsystem during the period of the failure). From time to time, during normal operation, each subsystem informs the other of the address ranges that it has hardened, whereupon the other subsystem removes these ranges from its record. Thus, upon recovery from a failure, the amount of data that must be copied back to the failed subsystem is

limited to the address ranges listed in the record maintained by the non-failed system, so that the time needed for full recovery is not too long.

Since data are recorded synchronously and records are maintained symmetrically on both the primary and secondary storage subsystems, the secondary storage subsystem can take the place of the primary storage subsystem immediately in case of a failure in the primary storage subsystem. Furthermore, read operations can be directed to either of the storage subsystems at any time.

{Claim summary will be inserted here in the final version.}

The present invention will be more fully understood from the following detailed description of the embodiments thereof, taken together with the drawings in which:

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram that schematically illustrates a data storage system, in accordance with an embodiment of the present invention;

Fig. 2 is a block diagram that schematically illustrates a storage subsystem, in accordance with an alternative embodiment of the present invention;

Fig. 3 is a schematic representation of bitmaps used in tracking data storage, in accordance with an embodiment of the present invention;

Fig. 4 is a flow chart that schematically illustrates a method for writing data to a data storage system, in accordance with an embodiment of the present invention; and

Fig. 5 is a flow chart that schematically illustrates a method for tracking data storage, in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF EMBODIMENTS

Fig. 1 is a block diagram that schematically illustrates a data storage system 20, in accordance with an embodiment of the present invention. System 20 comprises storage subsystems 22 and 24, which are labeled "storage node A" and storage node B" for convenience. In the description that follows, it is assumed that node A is configured as the primary storage subsystem, while node B is configured as the secondary storage subsystem for purposes of data mirroring. Thus, to write and read data to and from system 20, a host computer 26 (referred to hereinafter simply as a "host") communicates over a communication link 28 with subsystem 22. Typically, link 28 is part of a computer network, such as a storage area network (SAN). Alternatively, host 26 may communicate with subsystem 22 over substantially any suitable type of serial or parallel communication link. Although for the sake of simplicity, only a single host is shown in Fig. 1, system 20 typically serves multiple hosts. Typically, in normal operation, hosts may write data only to primary storage subsystem 22, but may read data from either subsystem 22 or 24.

Subsystems 22 and 24 may likewise comprise substantially any suitable type of storage device known in the art, such as a storage server, SAN disk device or network-attached storage (NAS) device. **{Is there any particular kind of storage device that you consider to be optimal for use in a system like this?}** Subsystems 22 and 24 may even comprise computer workstations, which are configured and programmed to carry out the storage functions described herein. Subsystems 22 and 24 may be

collocated in a single facility or, for enhanced data security, they may be located at mutually-remote sites. Although system 20 is shown in Fig. 1 as comprising only a single primary storage subsystem and a single secondary storage subsystems, the principles of the present invention may be applied in a straightforward manner to systems having greater numbers of primary and/or secondary storage subsystems. For example, the methods described hereinbelow may be extended to a system in which data written to a primary storage subsystem are mirrored on two different secondary storage subsystems in order to protect against simultaneous failures at two different points.

Each of subsystems 22 and 24 comprises a control unit (CU) 30, typically comprising one or more microprocessors, with a cache 32 and non-volatile storage media 34. Typically, cache 32 comprises volatile random-access memory (RAM), while storage media 34 comprise a magnetic disk or disk array. Alternatively, other types of volatile and non-volatile media may be used to carry out the cache and storage functions of subsystems 22 and 24. Control units 30 typically carry out the operations described herein under the control of software, which may be downloaded to subsystems 22 and 24 in electronic form, over a network, for example, or may be provided, alternatively or additionally, on tangible media, such as CD-ROM. Subsystems 22 and 24 communicate between themselves over a high-speed communication link 36, which may be part of a SAN or other network, or may alternatively be a dedicated line between the two subsystems. Subsystem 24 may also be coupled to communicate with host 26, as well as with other hosts

(not shown), over a communication link 38, similar to link 28. Link 38 enables subsystem 24 to serve as the primary storage subsystem in the event of a failure in subsystem 22.

Fig. 2 is a block diagram that schematically illustrates a storage subsystem 40, in accordance with an alternative embodiment of the present invention. Subsystem 40 may be used, for example, in place of storage subsystem 24 in system 20 (Fig. 1). Subsystem 40 is a sort of virtual storage node, made up of a local controller 42 and a remote disk 46. Controller 42 comprises CU 30 and memory 32, similar to the CU and memory used in subsystems 22 and 24. Disk 46, however, is not connected directly to CU 30, but instead communicates with the CU via a network 44. (For this purpose, disk 46 typically comprises a communication controller, not shown in the figures.) In this configuration, CU 30 may write and read data to and from disk 46 using a suitable network protocol, such as iSCSI, as is known in the art. The configuration of subsystem 40 is advantageous in that it allows the control units of the primary and second storage subsystems to be located at the same site, while disk 46 is located at a remote site. This arrangement facilitates rapid communication between the control units (thus reducing the latency of the data writing protocol described hereinbelow), while keeping backup data in disk 46 at a safe distance in case of a disaster at the primary site.

Fig. 3 schematically shows bitmaps 50 and 52, which are used by CU 30 in each of subsystems 22 and 24 for recording changes in the data stored by the other subsystem, in accordance with an embodiment of the

present invention. The use of these bitmaps is described hereinbelow in detail with reference to Figs. 4 and 5. Briefly, each bitmap 50, 52 comprises multiple bits 54, each corresponding to a storage element on disk 34. For example, each bit may correspond to a different track on the disk, or to some larger or smaller range of physical addresses on the disk. Certain bits 56 are marked by the CU (i.e., the bits are set in the bitmap) in each of subsystems 22 and 24 to indicate that data have been written to the cache in the other subsystem prior to transfer of the data to the corresponding storage elements on the disk. Alternatively, other types of data structures, as are known in the art, may be used for maintaining records of the status of data in caches 32.

Each CU 30 subsequently clears the marked bits 56 in its bitmap 50 or 52 when the CU is notified, as described hereinbelow, that the data have been transferred from cache 32 to disk 34 on the other storage subsystem. This process of transferring data from cache to disk may also be referred to as "hardening" or "destaging" the data. (Typically, "hardening" refers to any transfer of data from cache to disk, while "destaging" refers to processes whereby the CU performs an orderly transfer to disk of all the data in the cache or in a range of the cache.) Although the process of transferring data to disk may be applied to the entire cache at once - whereupon the CU clears the entire bitmap when the process is completed - it may be more efficient to apply the process to smaller ranges of addresses (i.e., smaller groups of tracks or other storage elements) on the disk. For this purpose, each of bitmaps 50 and 52 is divided into ranges 58, 60, 62 and 64. Each range is effectively treated as a

separate cache for purposes of tracking data transfer to disk. For each range, one of bitmaps 50 and 52 is treated as the current bitmap, in which CU 30 marks the appropriate bits when data are written to cache on the other subsystem, while the other bitmap is treated as the old bitmap, as described below. Although four ranges are shown in Fig. 3, cache 32 may alternatively be divided into a larger or smaller number of ranges for these purposes.

Fig. 4 is a flow chart that schematically illustrates a method used in writing data from host 26 to storage system 20, in accordance with an embodiment of the present invention. The method is invoked when host 26 writes data over link 28 to the primary storage subsystem, i.e., subsystem 22 in the present example, at a host writing step 60. Upon receiving the data, CU 30 of subsystem 22 places the data in cache 32, at a data caching step 62. CU 30 determines the track or tracks in which the data are to be stored on disks 34 in subsystems 22 and 24, and marks the corresponding bits 54 in the current bitmap 50 or 52. (As noted above, the bitmaps are just one example of a data structure that can be used to keep a record of the cache status, and each bit may alternatively correspond to a data element that is larger or smaller than a single track on the disk.) CU 30 of subsystem 22 then writes the data to subsystem 24 via link 36, at a data copying step 64.

CU 30 of secondary storage subsystem 24 receives the data over link 36, at a secondary receiving step 66. The CU of subsystem 24 places the data in its cache 32, and marks the bits in its bitmap 50 or 52 that correspond to the tracks for which the data are destined. Marked bits

56 in the bitmap held by secondary storage subsystem 24 indicate that primary storage subsystem 22 may have data in its cache that have not yet been written to the corresponding tracks on disk 34 of subsystem 22. After writing the data to cache 32, CU 30 of subsystem 24 sends an acknowledgment over link 36 to subsystem 22. Upon receiving the acknowledgment, CU 30 of subsystem 22 signals host 26, at an acknowledgment step 26, to acknowledge to the host operating system that the write operation was successfully completed. The acknowledgment is issued to host 26 independently of operations carried out on subsystems 22 and 24 to store the cached data to disks 34. Thus, the acknowledgment may typically be issued while the data are still in the volatile cache and before the data have actually been stored on disks 34 or any other non-volatile media.

Once data have been written to cache 32, each CU 30 proceeds to transfer the data to disk 34. After a given track or range of tracks has been hardened in this manner on one of the storage subsystems, the CU notifies the other storage subsystem, which then clears the corresponding bits in its old bitmap. The notification preferably refers to a range of tracks, rather than just a single track, since sending notifications too frequently creates substantial overhead traffic on link 36 between subsystems 22 and 24. Some methods that can be used to perform data hardening and to convey these "hardening notifications" efficiently are described hereinbelow. When the CU of one subsystem is notified that a given track has been hardened on the other subsystem, it clears the corresponding marked bits 56 on the old bitmap. In the meanwhile, as the CU receives new

write data (at step 62 or 66 above), it marks the corresponding bits in the current bitmap. A logical "OR" of the current and old bitmaps held by the CU in each of subsystems 22 and 24 then gives a map of all the tracks containing data that may not yet have been hardened on the other subsystem.

Fig. 5 is a flow chart that schematically illustrates one method for tracking data hardening in system 20, in accordance with an embodiment of the present invention. This method is based on destaging, whereby CU 30 of secondary subsystem 24 periodically scans its cache 32 (or scans a part of the cache corresponding to one of range 58, 60, 62 or 64) and writes all unhardened data to disk 34. Secondary storage subsystem 24 notifies primary storage subsystem 22 as the secondary subsystem destages each range. The identical method may be used to notify the secondary subsystem of destaging on the primary subsystem. Typically, the destage operation takes place at predetermined intervals or, alternatively or additionally, when CU 30 determines that the amount of unhardened data in a certain range of the cache (which may include the entire cache) is greater than some predetermined threshold. Note that in between these destaging operations, CU 30 may continue hardening data intermittently according to other criteria, as is known in the art.

Before beginning the destaging scan, CU 30 of subsystem 24 sends a message over link 36 to subsystem 22 to indicate that the scan has started, at a starting step 70. The message indicates the range of the cache that is to be destaged. By way of example, let us assume that the destaging operation is to be applied to range 58.

The range may alternatively include the entire cache. Upon receiving the message, CU 30 of subsystem 22 saves its current bitmap of range 58 (in which it has marked the tracks for which data have been written to subsystem 24 up to this point) as the old bitmap of range 58, at an old bitmap saving step 72. Referring to Fig. 3, let us assume that bitmap 50 has been in use up to this point as the current bitmap for range 58, and includes marked bits 56. Range 58 of bitmap 50 is now saved as the old bitmap. Any previously-saved old bitmap of range 58 is discarded. **{Why is this even necessary? Wouldn't any previously-saved old bitmap have been cleared upon completion of the last destage? On the other hand, if there are still bits marked in the old bitmap, it would seem to indicate that there are still unhardened data - maybe due to some failure in the last destaging run. Perhaps the old-old bitmap be OR-ed with the new-old bitmap.}** From this point forth, CU 30 of subsystem 22 uses bitmap 52 as the current bitmap for range 58, so that any tracks to which new data are written to cache in range 58 will now be recorded in bitmap 52. CU 30 of subsystem 22 then returns an acknowledgment to subsystem 24, at an acknowledgment step 74.

Upon receiving the acknowledgment, CU 30 of subsystem 24 begins its destaging scan of range 58, at a destaging step 76. When destaging of the entire range is finished, CU 30 of subsystem 24 sends another message to subsystem 22, indicating that the scan has been completed, at a completion message step 78. Upon receiving the message, CU 30 of subsystem 22 clears all

the bits 54 in range 58 of bitmap 50 (the old bitmap), at a bitmap clearing step 80.

Range 64 in Fig. 3 shows an example of an old bitmap range that has been cleared in bitmap 50, following which new bits 56 are marked in bitmap 52. As another example, in range 62, a destaging scan has started with respect to old bitmap 50, but has not yet been completed, so that some bits in range 62 of bitmap 50 are still marked. Meanwhile, as new data are written during the destaging scan, CU 30 of subsystem 22 has begun to mark bits in range 62 of the new current bitmap 52. Although in these examples, for the sake of clarity and convenience, bitmap 50 is referred to as the old bitmap, while bitmap 52 is referred to as the current bitmap, in actual operation the roles of "old" and "current" bitmap toggle back and forth between the two bitmaps.

To illustrate failure recovery in system 20, let us assume that subsystem 22 has failed, while subsystem 24 remains operational. At the time of failure, CU 30 of subsystem 24 held bitmaps 50 and 52. The union (logical OR) of all the bits that are marked in the two bitmaps indicates all the tracks of data in cache 32 of subsystem 22 that may have contained unhardened data at the time of failure. In fact, some of these tracks may already have been hardened, although notification did not reach subsystem 24. It can be said with certainty, however, that there are no tracks that have not been hardened on subsystem 22 whose corresponding bits are not marked in the union of bitmaps 50 and 52 held by CU 30 on subsystem 24. In other words, the union of these bitmaps represents a superset of all the unhardened tracks on subsystem 22.

At the time of failure, system 20 may "failover" to subsystem 24, so that subsystem 24 now serves as the primary (and only) storage subsystem. In this case, CU 30 of subsystem 24 maintains a further record, typically by marking additional bits in the united bitmap, indicating the tracks to which data are written while subsystem 22 is out of service.

When subsystem 22 is ready to return to service, CU 30 in subsystem 22 performs initial machine loading, as is known in the art, and then asks subsystem 24 for a data update. CU 30 of subsystem 24 then transfers to subsystem 22 the data in all the tracks that are marked in the united bitmap. Once the transfer is complete, subsystem 22 may resume operation as the primary storage subsystem.

Alternatively, other methods may be used for clearing bits in bitmaps 50 and 52, besides the periodic destaging method shown in Fig. 5. To exemplify one such method, we again consider tracking of data hardening on subsystem 24 (although the method may likewise be applied to hardening of data on subsystem 22). Control units 30 on both of subsystems 22 and 24 maintain similar bitmaps 50 and 52 with respect to the data tracks that have been copied from subsystem 22 to subsystem 24. Again taking region 58 as an example, as subsystem 22 conveys data to subsystem 24, both subsystems mark bits 56 in region 58 of bitmap 50 to indicate the tracks that are to be hardened. Subsystem 24 increments a counter N for each new bit that is marked in range 58 (and similarly in ranges 60, 62 and 64). When subsystem 24 hardens a track in region 58, it decrements N, without notifying subsystem 22. Subsystem 24 may choose the tracks to

harden using any suitable criterion, such as hardening least-recently-used tracks. There is no need for subsystem 24 to perform an orderly destaging of an entire region, as in the method of Fig. 5.

{I found the explanation in the disclosure of the counters N and M a little obscure. I tried to phrase it here in a way that is (I hope) more transparent, based on the explanation you gave me at our meeting. Have I done it correctly?}

Periodically, subsystem 24 sends a message to subsystem 22 to indicate that it is about to switch to a new bitmap for a given region, say region 58, and waits for subsystem 22 to acknowledge the message. Region 58 of bitmap 50 is then locked in both subsystems 22 and 24, and all subsequent data writes to the region are marked in bitmap 52. A new counter N for region 58 in bitmap 52 is set initially to zero and is then incremented and decremented as described above. Subsystem 24 meanwhile continues to harden the tracks that are marked in region 58 of bitmap 50 as containing cached data that are yet to be hardened. Subsystem 24 decrements N for each track that it hardens in the old bitmap. When N reaches zero, subsystem 24 notifies subsystem 22 that all tracks in region 58 of bitmap 50 have been hardened, whereupon subsystem 22 clears all the bits in region 58 of old bitmap 50, as at step 80 in the method of Fig. 5. The process toggles back and forth between bitmaps 50 and 52, as described above.

As another alternative, subsystem 22 may scan a given region in the old bitmap, indicating unhardened tracks on subsystem 24, and may compare the result to the

bitmap of unhardened tracks in its own local cache 32. Subsystem 22 may then signal subsystem 24 to perform a synchronous destaging of all tracks that were found to contain unhardened data in both the old bitmap and the local cache. **{This was described only very briefly in your disclosure, and we did not discuss it. Did I understand it correctly? I am not sure why it should work. What happens to tracks that are clean in the local cache but are not yet destaged at the other end?}**

As noted above, although certain configurations of system 20 and certain particular data mirroring protocols are described above in order to illustrate the principles of the present invention, these principles may similarly be applied in other system configurations and using other protocols, as will be apparent to those skilled in the art. It will thus be appreciated that the embodiments described above are cited by way of example, and that the present invention is not limited to what has been particularly shown and described hereinabove. Rather, the scope of the present invention includes both combinations and subcombinations of the various features described hereinabove, as well as variations and modifications thereof which would occur to persons skilled in the art upon reading the foregoing description and which are not disclosed in the prior art.

CLAIMS

1. A method for storing data in a data storage system that includes primary and secondary storage subsystems, including respective first and second volatile cache memories and respective first and second non-volatile storage media, the method comprising:

receiving the data at the primary storage subsystem from a host processor;

writing the data to the first volatile cache memory in the primary storage subsystem;

copying the data from the primary storage subsystem to the secondary storage subsystem;

writing the copied data to the second volatile cache memory in the secondary storage subsystem;

returning an acknowledgment from the secondary storage subsystem to the primary storage subsystem responsively to writing the copied data to the second volatile cache memory and prior to saving the data in the second non-volatile storage media;

signaling the host processor that the data have been stored in the data storage system responsively to the acknowledgment from the secondary storage subsystem; and

transferring the data in the primary and second storage subsystems from the first and second volatile cache memories to the first and second non-volatile storage media, respectively.

2. The method according to claim 1, wherein copying the data comprises transmitting the data between mutually-remote sites over a communication link between the sites.

3. The method according to claim 1, wherein the second volatile cache memory and the second non-volatile storage

media are located in mutually-remote sites, and wherein transferring the data comprises transmitting the data from the second volatile cache memory to the second non-volatile storage media over a communication link between the sites.

4. The method according to claim 1, wherein copying the data comprises creating a mirror on the secondary storage subsystem of the data received by the primary storage subsystem.

5. The method according to claim 4, and comprising, upon occurrence of a failure in the primary storage subsystem, configuring the secondary storage subsystem to serve as the primary storage subsystem so as to receive further data from the host processor to be stored by the data storage system.

6. The method according to claim 1, wherein transferring the data comprises sending a message from the secondary storage subsystem to the primary storage subsystem indicating addresses of the data that have been transferred to the second non-volatile storage media, and wherein the method further comprises creating a record on the primary storage subsystem of the addresses of the data copied to the secondary storage subsystem, and updating the record in response to the message.

7. The method according to claim 6, and further comprising, upon recovery of the system from a failure of the second storage subsystem, conveying, responsively to the record, a portion of the data from the primary storage subsystem to be stored on the secondary storage subsystem.

8. The method according to claim 7, wherein updating the record comprises removing from the record the addresses of the data that have been transferred to the second non-volatile storage media.

9. The method according to claim 6, wherein creating the record comprises marking respective bits in a bitmap corresponding to addresses of the data copied to the secondary storage subsystem, and wherein updating the record comprises clearing the respective bits.

10. The method according to claim 6, wherein transferring the data comprises transferring the data in a range of the addresses from the second volatile cache memory to the second non-volatile storage media, and wherein sending the message comprises informing the primary storage subsystem that the data in the range have been transferred, so that the primary storage subsystem updates the record with respect to the range.

11. The method according to claim 10, wherein transferring the data in the range comprises destaging the range of the addresses.

12. The method according to claim 1, wherein transferring the data comprises sending a message from the primary storage subsystem to the secondary storage subsystem indicating addresses of the data that have been transferred to the first non-volatile storage media, and wherein the method further comprises creating a record on the secondary storage subsystem of the addresses of the data copied to the secondary storage subsystem, and updating the record in response to the message.

13. The method according to claim 12, and further comprising, upon recovery of the system from a failure of the primary storage subsystem, conveying, responsively to the record, a portion of the data from the secondary storage subsystem to be stored on the primary storage subsystem.

14. The method according to claim 13, wherein updating the record comprises removing from the record the addresses of the data that have been transferred to the primary non-volatile storage media.

15. The method according to claim 12, wherein creating the record comprises marking respective bits in a bitmap corresponding to addresses of the data copied to the secondary storage subsystem, and wherein updating the record comprises clearing the respective bits.

16. The method according to claim 12, wherein transferring the data comprises transferring the data in a range of the addresses from the first volatile cache memory to the first non-volatile storage media, and wherein sending the message comprises informing the secondary storage subsystem that the data in the range have been transferred, so that the secondary storage subsystem updates the record with respect to the range.

17. The method according to claim 16, wherein transferring the data in the range comprises destaging the range of the addresses.

18. A data storage system, comprising:

a primary storage subsystem, which comprises a first volatile cache memory and first non-volatile storage media; and

a secondary storage subsystem, which comprises a second volatile cache memory and second non-volatile storage media,

wherein the primary storage subsystem is arranged to receive data from a host processor, to write the data to the first volatile cache memory, to copy the data to the secondary storage subsystem, and to transfer the data from the first volatile cache memory to the first non-volatile storage media, and

wherein the second storage subsystem is arranged to receive and write the copied data to the second volatile cache memory, to transfer the data from the first volatile cache memory to the first non-volatile storage media, and to return an acknowledgment to the primary storage subsystem responsively to writing the copied data to the second volatile cache memory and prior to transferring the data to the second non-volatile storage media,

wherein the primary storage subsystem is arranged to signal the host processor that the data have been stored in the data storage system responsively to the acknowledgment from the secondary storage subsystem.

19. The system according to claim 18, wherein the primary and secondary storage subsystems are located at mutually-remote sites, and are coupled to communicate over a communication link between the sites.

20. The system according to claim 18, wherein the second volatile cache memory and the second non-volatile storage media are located in mutually-remote sites, and wherein the second storage subsystem is arranged to transfer the data from the second volatile cache memory to the second

non-volatile storage media over a communication link between the sites.

21. The system according to claim 18, wherein the secondary storage subsystem is arranged to mirror the data held by the primary storage subsystem.

22. The system according to claim 21, wherein the secondary storage subsystem is configurable, upon occurrence of a failure in the primary storage subsystem, to serve as the primary storage subsystem so as to receive further data from the host processor to be stored by the data storage system.

23. The system according to claim 18, wherein the secondary storage subsystem is arranged to send a message to the primary storage subsystem indicating addresses of the data that have been transferred to the second non-volatile storage media, and wherein the primary storage subsystem is arranged to create a record of the addresses of the data copied to the secondary storage subsystem, and to update the record in response to the message.

24. The system according to claim 23, wherein the primary storage subsystem is arranged, upon recovery of the system from a failure of the second storage subsystem, to convey, responsively to the record, a portion of the data from the primary storage subsystem to be stored on the secondary storage subsystem.

25. The system according to claim 24, wherein the primary storage subsystem is arranged to remove from the record the addresses of the data that have been transferred to the second non-volatile storage media.

26. The system according to claim 23, wherein the primary storage subsystem is arranged to create the record by marking respective bits in a bitmap corresponding to addresses of the data copied to the secondary storage subsystem, and to update the record by clearing the respective bits in response to the message.

27. The system according to claim 23, wherein the secondary storage subsystem is arranged to transfer the data over a range of the addresses from the second volatile cache memory to the second non-volatile storage media, and to indicate in the message that the data in the range have been transferred, so that the primary storage subsystem updates the record with respect to the range.

28. The system according to claim 27, wherein the secondary storage subsystem is arranged to transfer the data by destaging the range of the addresses.

29. The system according to claim 18, wherein the primary storage subsystem is arranged to send a message to the secondary storage subsystem indicating addresses of the data that have been transferred to the first non-volatile storage media, and wherein the secondary storage subsystem is arranged to create a record of the addresses of the data copied to the secondary storage subsystem, and to update the record in response to the message.

30. The system according to claim 29, wherein the secondary storage subsystem is arranged, upon recovery of the system from a failure of the primary storage subsystem, to convey, responsively to the record, a

portion of the data from the secondary storage subsystem to be stored on the primary storage subsystem.

31. The system according to claim 30, wherein the secondary storage subsystem is arranged to remove from the record the addresses of the data that have been transferred to the first non-volatile storage media.

32. The system according to claim 29, wherein the secondary storage subsystem is arranged to create the record by marking respective bits in a bitmap corresponding to addresses of the data copied to the secondary storage subsystem, and to update the record by clearing the respective bits in response to the message.

33. The system according to claim 29, wherein the primary storage subsystem is arranged to transfer the data over a range of the addresses from the first volatile cache memory to the first non-volatile storage media, and to indicate in the message that the data in the range have been transferred, so that the secondary storage subsystem updates the record with respect to the range.

34. The system according to claim 33, wherein the primary storage subsystem is arranged to transfer the data by destaging the range of the addresses.

35. A computer software product for use in a data storage system including primary and secondary storage subsystems, which include respective first and second control units, respective first and second volatile cache memories, and respective first and second non-volatile storage media, the product comprising a computer-readable medium in which program instructions are stored, which

instructions, when read by the first and second control units, cause the first control unit to receive data from a host processor, to write the data to the first volatile cache memory, to copy the data to the secondary storage subsystem, and to transfer the data from the first volatile cache memory to the first non-volatile storage media, and cause the second control unit to receive and write the copied data to the second volatile cache memory, to transfer the data from the first volatile cache memory to the first non-volatile storage media, and prior to transferring the data to the second non-volatile storage media, to return an acknowledgment to the primary storage subsystem responsively to writing the copied data to the second volatile cache memory, wherein the instructions further cause the first control unit to signal the host processor that the data have been stored in the data storage system responsively to the acknowledgment from the secondary storage subsystem.

36. The product according to claim 35, wherein the primary and secondary storage subsystems are located at mutually-remote sites, and wherein the instructions cause the first and second control units to communicate over a communication link between the sites.

37. The product according to claim 35, wherein the second volatile cache memory and the second non-volatile storage media are located in mutually-remote sites, and wherein the instructions cause the second control unit to transfer the data from the second volatile cache memory to the second non-volatile storage media over a communication link between the sites.

38. The product according to claim 35, wherein the instructions cause the first and second control units to mirror the data held by the primary storage subsystem on the secondary storage subsystem.

39. The product according to claim 38, wherein the instructions cause the secondary storage subsystem, upon occurrence of a failure in the primary storage subsystem, to serve as the primary storage subsystem so as to receive further data from the host processor to be stored by the data storage system.

40. The product according to claim 35, wherein the instructions cause the second control unit to send a message to the primary storage subsystem indicating addresses of the data that have been transferred to the second non-volatile storage media, and further cause the first control unit to create a record of the addresses of the data copied to the secondary storage subsystem, and to update the record in response to the message.

41. The product according to claim 40, wherein the instructions cause the first control unit, upon recovery of the system from a failure of the second storage subsystem, to convey, responsively to the record, a portion of the data from the primary storage subsystem to be stored on the secondary storage subsystem.

42. The product according to claim 41, wherein the instructions cause the first control unit to remove from the record the addresses of the data that have been transferred to the second non-volatile storage media.

43. The product according to claim 40, wherein the instructions cause the first control unit to create the

record by marking respective bits in a bitmap corresponding to addresses of the data copied to the secondary storage subsystem, and to update the record by clearing the respective bits in response to the message.

44. The product according to claim 40, wherein the instructions cause the second control unit to transfer the data over a range of the addresses from the second volatile cache memory to the second non-volatile storage media, and to indicate in the message that the data in the range have been transferred, so that the first control unit updates the record with respect to the range.

45. The product according to claim 44, wherein the instructions cause the second control unit to transfer the data by destaging the range of the addresses.

46. The product according to claim 35, wherein the instructions cause the first control unit to send a message to the secondary storage subsystem indicating addresses of the data that have been transferred to the first non-volatile storage media, and further cause the second control unit to create a record of the addresses of the data copied to the secondary storage subsystem, and to update the record in response to the message.

47. The product according to claim 46, wherein the instructions cause the second control unit, upon recovery of the system from a failure of the primary storage subsystem, to convey, responsively to the record, a portion of the data from the secondary storage subsystem to be stored on the primary storage subsystem.

48. The product according to claim 47, wherein the instructions cause the second control unit to remove from the record the addresses of the data that have been transferred to the first non-volatile storage media.

49. The product according to claim 46, wherein the instructions cause the second control unit to create the record by marking respective bits in a bitmap corresponding to addresses of the data copied to the secondary storage subsystem, and to update the record by clearing the respective bits in response to the message.

50. The product according to claim 46, wherein the instructions cause the first control unit to transfer the data over a range of the addresses from the first volatile cache memory to the first non-volatile storage media, and to indicate in the message that the data in the range have been transferred, so that the secondary storage subsystem updates the record with respect to the range.

51. The product according to claim 50, wherein the instructions cause the first control unit to transfer the data by destaging the range of the addresses.

EXHIBIT H

From: "Aviad Zlotnick" <AVIAD@il.ibm.com>
To: "Daniel Kligler" <dkligler@stc.co.il>
Date: 8/10/03 1:28PM
Subject: Re: IL9-2003-0033 (Low-cost PPRC) - our ref. 49267 - IBM CONFIDENTIAL

Danny,

I've read the first draft. I want to clarify the three methods of maintaining the CR bitmaps. In all three cases we have an old CR bitmap and a new one. Each site needs to tell the other site when the old bitmap can be disregarded. Usually this is also a good time to toggle bitmaps.

In the first method we rely on the control unit's cache scan. This scan goes through the whole cache, including the cache for devices that at not in any PPRC relation, and destages the dirty cache entries. When the scan is done we are guaranteed that the tracks that are marked in the remote's old bitmap are hardened. I think you got this one perfectly.

My description of the second and third methods was incomplete. I think you noticed that. Anyway, in both these methods the local sit needs a copy of the remote site's bitmaps. This is easy to obtain, since the local site controls the toggle of the bitmaps on the remote. The second and third methods both use on the local copy of the remote bitmaps.

In the second method we use a counter to know when the tracks marked in the old bitmap have been hardened. First, when the old bitmap first comes into existence, count the number of set bits in it. Let the count be M. Now, whenever there is a new write to a track that is dirty in cache, and is not yet marked in the new CR bitmap, but is marked in the old one, decrement M. Also, when the CU destages a track that is marked in the old bitmap but not in the new one, decrement M. When M becomes 0 we are sure that the remote site no longer needs its old CR bitmap.

The third method is similar to the first, except that instead of modifying the CUs cache scan we perform a special cache scan that scans only the tracks that are marked in the old bitmap.

Regarding the Seneca paper:

The volatile immediate report mode acknowledges the write as soon as it is written in either cache, not both. That is why it is sensitive to a single point of failure, and that is why it is not a problem for our invention. The bidirectional mirroring mentioned in the Seneca paper relates to different sets of data (LUNs), and so it is not a problem for SAM (IL9-2003-0028).

Finally, regarding your question about the bitmap that marks which tracks are not hardened locally - cache usually uses a different data structure, but it is, as you thought, independent of the CR bitmaps we are describing.

Additional remarks about the draft:
Page 6, line 26: the optimal kind of storage for this low end, inexpensive solution is JBOD (Just a Bunch of Disks), that is simple disks. Of course, one could use a RAID configuration and get better quality.

Page 8, line 28: please add something about having the second cache in an especially secure place, maybe somewhat remote from the primary site, and that providing such a safe location with its hardware may be a service offering. This should also be reflected in the claims.

Page 13, line 11: discarding is done by toggling the bitmaps, and since the previous old bitmap now becomes the new one, it must be cleared. Note that the only thing that clears bits is a toggle command - the bits represent a state in the remote machine, so nothing local is allowed to touch the bits. (This is where I went wrong in the original counter based scheme).

The claims look OK.

Regards,

Aviad

Tel: +972-48-296-284, Cell: +972-66-976-284, Fax: +972-48-296-112, Email: aviad@il.ibm.com

"Daniel Kligler" <dkligler@stc.co.il>
07/15/2003 10:52 AM

To: Aviad Zlotnick/Haifa/IBM@IBMIL
cc: Suzanne Erez/Haifa/IBM@IBMIL
Subject: IL9-2003-0033 (Low-cost PPRC) - our ref. 49267 - IBM CONFIDENTIAL

Dear Aviad,

One of the ideas we considered claiming in this application is that the host receives a write acknowledge when both the primary and the secondary have written the data to their volatile caches, rather than to non-volatile memory or disk. Have I understood this point correctly? It appears that a related idea is mentioned in the article you gave me by Ji et al., on page 257 ("Volatile immediate-report"). Ji seems to be talking about acknowledging the write as soon as the data are written to the primary cache, but he also says, just prior to this paragraph, that "these observations can be applied at both the primary and secondary copies." Any comments on this point?

Also note on page 258 the mention of "bidirectional mirroring" - may be relevant to IL9-2003-0028. Do you have any more information on this point?

Otherwise, there does not seem to be much resemblance between Ji's "Seneca" protocol and your method.

Regards,
Danny

CC: "HRL IP Department" <HRL_IPDEPARTMENT@il.ibm.com>, "Suzanne Erez" <SUZANNE@il.ibm.com>

EXHIBIT I

From: Daniel Kligler
To: Zlotnick, Aviad
Date: 8/13/03 1:01PM
Subject: Re: IL9-2003-0033 (Low-cost PPRC) - our ref. 49267 - IBM CONFIDENTIAL

Dear Aviad,

Attached please find a revised draft of this application (text and figures), incorporating your comments and corrections. As I wrote you in my e-mail yesterday, I have added a figure and also claims on your "second and third methods." I have also added a claim, as you requested, on provided a storage service. In preparation for filing, I have added an abstract and paraphrased the claims in the Summary of the Invention.

Please review this draft, and let me have any further corrections and/or your approval to file the application.

Regards,
Danny

>>> "Aviad Zlotnick" <AVIAD@il.ibm.com> 08/10/03 12:27PM >>>
Danny,

I've read the first draft. I want to clarify the three methods of maintaining the CR bitmaps. In all three cases we have an old CR bitmap and a new one. Each site needs to tell the other site when the old bitmap can be disregarded. Usually this is also a good time to toggle bitmaps.

In the first method we rely on the control unit's cache scan. This scan goes through the whole cache, including the cache for devices that at not in any PPRC relation, and destages the dirty cache entries. When the scan is done we are guaranteed that the tracks that are marked in the remote's old bitmap are hardened. I think you got this one perfectly.

My description of the second and third methods was incomplete. I think you noticed that. Anyway, in both these methods the local sit needs a copy of the remote site's bitmaps. This is easy to obtain, since the local site controls the toggle of the bitmaps on the remote. The second and third methods both use on the local copy of the remote bitmaps.

In the second method we use a counter to know when the tracks marked in the old bitmap have been hardened. First, when the old bitmap first comes into existence, count the number of set bits in it. Let the count be M. Now, whenever there is a new write to a track that is dirty in cache, and is not yet marked in the new CR bitmap, but is marked in the old one, decrement M. Also, when the CU destages a track that is marked in the old bitmap but not in the new one, decrement M. When M becomes 0 we are sure that the remote site no longer needs its old CR bitmap.

The third method is similar to the first, except that instead of modifying the CUs cache scan we perform a special cache scan that scans only the tracks that are marked in the old bitmap.

Regarding the Seneca paper:

The volatile immediate report mode acknowledges the write as soon as it is written in either cache, not both. That is why it is sensitive to a single point of failure, and that is why it is not a problem for our invention. The bidirectional mirroring mentioned in the Seneca paper relates to different sets of data (LUNs), and so it is not a problem for SAM (IL9-2003-0028).

Finally, regarding your question about the bitmap that marks which tracks are not hardened locally - cache usually uses a different data structure, but it is, as you thought, independent of the CR bitmaps we are describing.

Additional remarks about the draft:
Page 6, line 26: the optimal kind of storage for this low end, inexpensive solution is JBOD (Just a Bunch of Disks), that is simple disks. Of course, one could use a RAID configuration and get better quality.

EXHIBIT J

From: "Aviad Zlotnick" <AVIAD@il.ibm.com>
To: "Daniel Kligler" <dkligler@stc.co.il>
Date: 8/27/03 1:34PM
Subject: Re: IL9-2003-0033, your ref. 49267S1 - IBM CONFIDENTIAL

Danny,

I read the second draft (except for the system and apparatus claims). I have the following comments:

Page 5, line 4, last word should be "secondary", not "second".

Page 7, after line 12: If I understood what you wrote, then on page 6 you described the "second" and "third" methods where the secondary tells the primary to discard the old bitmap, but you only describe the first method for the other direction. Is that OK?

Page 8, line 8 at the end. What does the 41 stand for?

Page 13 line 29 and more: a basic misunderstanding: In this configuration it is the second cache that should be bunkered, and the service is for that cache site. The remote disk is in a recovery site that is most likely controlled by the owner of the first site. The typical configuration in this case is a production site in one place, a bunker site, or relay site, about 100 KM away, and a recovery site hundreds or thousands of KM away.

Page 14, line 25 and more: I think what you call destaging is what we call a destage scan. Destaging is an operation of copying cache content to the disk. As a result, the data is hardened. A destage scan is an operation of scanning a part of the cache, and destaging those tracks that are dirty. I think this distinction has implications on the claims.

Page 17, line 3, 4th word: I think you meant a track range or a set of tracks, not a single track.

Page 17, line 15, last word: not "destaging" but a "destage scan".

Page 19, line 6: I don't think that the old bitmap should be modified before the toggle signal from the remote site. Maybe you meant that the destage scan of some sections of the cache has ended on the secondary and a toggle signal was sent to the primary, whereas the destage scan on other sections is still in progress (or has not even started).

Page 26, line 17: It is the volatile part of storage that is a good candidate for a service provider. See my remark regarding Page 13, line 29.

Page 27, line 27, last word, and Page 29, line 9, last word: I think this should be a destage scan.

Thank you,

Aviad

Tel: +972-48-296-284, Cell: +972-66-976-284, Fax: +972-48-296-112, Email: aviad@il.ibm.com

CC: "Suzanne Erez" <SUZANNE@il.ibm.com>

EXHIBIT K

From: Daniel Kligler
To: Zlotnick, Aviad
Date: 8/31/03 11:22AM
Subject: Re: IL9-2003-0033, our ref. 49267 - IBM CONFIDENTIAL

Dear Aviad,

Attached please find a revised draft of this application, incorporating most of your corrections and comments below.

Note that Word paginates differently on different computers (depending generally on the printer you have configured as your default), so that your page and line numbers are generally NOT the same as mine. I had some difficulty figuring out the references, particularly the one on "page 19, line 6" that you mention below. It is easier for me if you quote a phrase from my text in your comment, which I can then search for easily. As an alternative, the attached Zip file contains a pdf version of the text, in which the page and line numbers will be stable regardless of the computer used to view the file. Let me know your preference for future cases.

Regarding your comment on page 7, after line 12 - this is (I think) just paraphrasal of the independent apparatus claim. I believe this claim covers all the different methods. If you want to double-check, you should have a look at the claim itself.

Regarding the term "destaging," there is a definition I have written in the specification, indicating that destaging relates to performing a destage scan. I have edited the specification to replace occurrences of "destaging" with "destage scan" where appropriate. The references to destaging in the claims all refer to "destaging a range," which I think covers the concept very clearly. I therefore did not change the claims in this respect.

Regarding your comment on page 19, line 6 - I am guessing that you are referring here to range 62. What I am trying to show here is that the CU has "frozen" the range in old bitmap 50, while awaiting the completion of the destage scan on the other subsystem, and meanwhile has started to write to the same range in the new bitmap. Please look again at the explanation in the text. (Or maybe you are referring here to something else altogether.) Let me know if further correction is needed.

Please let me know if you have further corrections, or if this draft is now ready for filing.

Regards,
Danny

>>> "Aviad Zlotnick" <AVIAD@il.ibm.com> 08/27/03 12:37PM >>>

Danny,

I read the second draft (except for the system and apparatus claims). I have the following comments:

Page 5, line 4, last word should be "secondary", not "second".

Page 7, after line 12: If I understood what you wrote, then on page 6 you described the "second" and "third" methods where the secondary tells the primary to discard the old bitmap, but you only describe the first method for the other direction. Is that OK?

Page 8, line 8 at the end. What does the 41 stand for?

Page 13 line 29 and more: a basic misunderstanding: In this configuration it is the second cache that should be bunkered, and the service is for that cache site. The remote disk is in a recovery site that is most likely controlled by the owner of the first site. The typical configuration in this case is a production site in one place, a bunker site, or relay site, about 100 KM away, and a recovery site hundreds or thousands of KM away.

Page 14, line 25 and more: I think what you call destaging is what we call a destage scan. Destaging is an operation of copying cache content to the disk. As a result, the data is hardened. A destage scan is an operation of scanning a part of the cache, and destaging those tracks that are dirty. I think this distinction has implications on the claims.

Page 17, line 3, 4th word: I think you meant a track range or a set of tracks, not a single track.

EXHIBIT L

From: Daniel Kligler
To: Erez, Suzanne
Date: 9/1/03 2:23PM
Subject: Re: IL9-2003-0033, our ref. 49267 - IBM CONFIDENTIAL

Dear Suzanne,

Clean, final version attached.

The figures (on paper) will follow under separate cover.

Regards,
Daniel

>>> "Suzanne Erez" <SUZANNE@il.ibm.com> 09/01/03 12:41PM >>>

Daniel,
I thought I sent it, my apologies if I did not.
I will fax one immediately - to what fax number?

And then, yes, please send me the clean copies of -0033 for filing.

Suzanne

Suzanne Erez
IP Department
IBM Haifa Laboratories
Tel: 972-4-829-6069 Fax: 972-4-829-6521
suzanne@il.ibm.com

All human beings should try to learn before they die what they are running from, and to, and why. -James Thurber, writer and cartoonist (1894-1961)

"Daniel Kligler" <dkligler@stc.co.il> 01-09-03 01:30 PM

To: Suzanne Erez/Haifa/IBM@IBMIL
cc: "Ronen Harel" <RonenH@stc.co.il>
Subject: Re: IL9-2003-0033, our ref. 49267 - IBM CONFIDENTIAL

Dear Suzanne,

One further question in this regard - a while back we received a letter from you asking us to change the format of our formal drawings. We were not sure exactly what was required, and therefore asked you to give us a sample of the proper format. As far as I know, we have not yet received the sample. Can you provide it?

Thanks and regards,
Daniel

>>> "Aviad Zlotnick" <AVIAD@il.ibm.com> 09/01/03 11:18AM >>>
Danny,

0033 is now ready for filing. Thanks for your hard work!

Aviad

Tel: +972-48-296-284, Cell: +972-66-976-284, Fax: +972-48-296-112, Email: aviad@il.ibm.com

EXHIBIT M

From: "Suzanne Erez" <SUZANNE@il.ibm.com>
To: <dkligler@stc.co.il>
Date: 9/14/03 9:34AM
Subject: Fw: IL920030031US1, (49265) 32US1 (49266) and 33US1 (49267)

(See attached file: heading.doc)
Hello Daniel,
Pleas see the letter below. IBM Watson could not file the patent applications due to the figures. Please add to the header of each figure the initials SCK, as in the document above. The initials did appeared on the sample sent to Ronen, and perhaps he did not understand the significance.
Thanks
Suzanne

Suzanne Erez
IP Department
IBM Haifa Laboratories
Tel: 972-4-829-6069 Fax: 972-4-829-6521
suzanne@il.ibm.com

Any sufficiently advanced technology is indistinguishable from magic.
- Murphy's Technology Laws
----- Forwarded by Suzanne Erez/Haifa/IBM on 14-09-03 09:25 AM -----

Michelle Parra@IBMUS To: Suzanne Erez/Haifa/IBM@IBMIL@IBMDE
cc: Hadas Cohen Bar-Gil/Haifa/IBM@IBMIL, Stephen C Kaufman/Watson/IBM@IBMUS
11-09-03 04:55 PM From: Michelle Parra/Watson/IBM@IBM Research
Subject: Re: IL920030031US1, 32US1 and 33US1 (Document link: Suzanne Erez)

Hello Suzanne,

I am not able to edit the drawings for these three applications. Could you please send me an editable version or have the drawings corrected for me. Thank you.

Each drawings should have their respective docket numbers without dashes and with US1 if the application is original.
In the case that the application is a converted provisional application, then it should be US2.

IBM Guidelines: Heading for the Formal Drawings:

Heading:

Provide a three line heading at the top of the first page of the drawing not to exceed 2 3/4" (7.0 cm) in width, centered between side edges within 3/4" (19.1 mm) of the top edge. On the first line of the heading, indicate the sheet number of the drawing as for example, 1/3, 2/3, or 3/3. On the second line, list the first inventor's last name with "et al", if necessary. On the third line, list the IBM docket number and IBM attorney initials. On subsequent pages indicate sheet number and IBM docket number only.

On the page the heading should look like this and situated no lower than the appended sample:

EXHIBIT M

1/5
Tross et al.
IL920030033US1 SCK